package com.twitter.search.common.relevance.features;

import java.util.Collection;

import java.util.List;

import java.util.Set;

import com.google.common.annotations.VisibleForTesting;

import com.google.common.collect.Sets;

import com.twitter.common.text.token.TokenizedCharSequence;

public class TweetTextFeatures {

// Basic Features, always extracted.

// normalized, lower cased tweet text, w/o resolved urls

private String normalizedText;

// tokens from normalizedText, w/o resolved urls, lower cased.

private List<String> tokens;

// tokens from resolved urls, lower cased.

private List<String> resolvedUrlsTokens;

// tokens in the form of a TokenizedCharSeq, NOT LOWER CASED

private TokenizedCharSequence tokenSequence;

// strippedTokens above joined with space

private String normalizedStrippedText;

// normalized, original case tokens, without @mention, #hashtag or urls.

private List<String> strippedTokens;

// all hash tags, without "#", lower cased

private Set<String> hashtags = Sets.newHashSet();

// all mentions, without "@", lower cased

private Set<String> mentions = Sets.newHashSet();

// whether this tweet has a question mark that's not in url.

private boolean hasQuestionMark = false;

private boolean hasPositiveSmiley = false;

private boolean hasNegativeSmiley = false;

// normalized, original case smileys

private List<String> smileys;

// lower cased, normalized stock names, without "$"

private List<String> stocks;

// Extra features for text quality evaluation only.

private int signature = TweetIntegerShingleSignature.DEFAULT\_NO\_SIGNATURE;

private Set<String> trendingTerms = Sets.newHashSet();

private int length;

private int caps;

public String getNormalizedText() {

return normalizedText;

}

public void setNormalizedText(String normalizedText) {

this.normalizedText = normalizedText;

}

public List<String> getTokens() {

return tokens;

}

public int getTokensSize() {

return tokens == null ? 0 : tokens.size();

}

public void setTokens(List<String> tokens) {

this.tokens = tokens;

}

public List<String> getResolvedUrlTokens() {

return resolvedUrlsTokens;

}

public int getResolvedUrlTokensSize() {

return resolvedUrlsTokens == null ? 0 : resolvedUrlsTokens.size();

}

public void setResolvedUrlTokens(List<String> tokensResolvedUrls) {

this.resolvedUrlsTokens = tokensResolvedUrls;

}

public TokenizedCharSequence getTokenSequence() {

return tokenSequence;

}

public void setTokenSequence(TokenizedCharSequence tokenSequence) {

this.tokenSequence = tokenSequence;

}

public String getNormalizedStrippedText() {

return normalizedStrippedText;

}

public void setNormalizedStrippedText(String normalizedStrippedText) {

this.normalizedStrippedText = normalizedStrippedText;

}

public List<String> getStrippedTokens() {

return strippedTokens;

}

public int getStrippedTokensSize() {

return strippedTokens == null ? 0 : strippedTokens.size();

}

public void setStrippedTokens(List<String> strippedTokens) {

this.strippedTokens = strippedTokens;

}

public Set<String> getHashtags() {

return hashtags;

}

public int getHashtagsSize() {

return hashtags.size();

}

public void setHashtags(Collection<String> hashtags) {

this.hashtags = Sets.newHashSet(hashtags);

}

public Set<String> getMentions() {

return mentions;

}

public int getMentionsSize() {

return mentions.size();

}

public void setMentions(Collection<String> mentions) {

this.mentions = Sets.newHashSet(mentions);

}

public boolean hasQuestionMark() {

return hasQuestionMark;

}

public void setHasQuestionMark(boolean hasQuestionMark) {

this.hasQuestionMark = hasQuestionMark;

}

public boolean hasPositiveSmiley() {

return hasPositiveSmiley;

}

public void setHasPositiveSmiley(boolean hasPositiveSmiley) {

this.hasPositiveSmiley = hasPositiveSmiley;

}

public boolean hasNegativeSmiley() {

return hasNegativeSmiley;

}

public void setHasNegativeSmiley(boolean hasNegativeSmiley) {

this.hasNegativeSmiley = hasNegativeSmiley;

}

public List<String> getSmileys() {

return smileys;

}

public int getSmileysSize() {

return smileys == null ? 0 : smileys.size();

}

public void setSmileys(List<String> smileys) {

this.smileys = smileys;

}

public List<String> getStocks() {

return stocks;

}

public int getStocksSize() {

return stocks == null ? 0 : stocks.size();

}

public void setStocks(List<String> stocks) {

this.stocks = stocks;

}

public int getSignature() {

return signature;

}

public void setSignature(int signature) {

this.signature = signature;

}

/\*\* Returns the trending terms. \*/

public Set<String> getTrendingTerms() {

return trendingTerms;

}

public int getTrendingTermsSize() {

return trendingTerms.size();

}

@VisibleForTesting

public void setTrendingTerms(Set<String> trendingTerms) {

this.trendingTerms = trendingTerms;

}

public int getLength() {

return length;

}

public void setLength(int length) {

this.length = length;

}

public int getCaps() {

return caps;

}

public void setCaps(int caps) {

this.caps = caps;

}

}